音 楽 情 報 科 学 49−14
（２００３．２．２２）

ユーザの音楽嗜好に基づく音楽情報検索手法

帆足 啓一郎　　井ノ上 直己

KDDI 研究所

〒 356-8502 上福岡市大原 2-1-15

E-mail: {hoashi,inoue}@kddilabs.jp

本研究では，ユーザから提供される少量の学習データから，ユーザの音楽的嗜好を抽出し，ユーザが好む音楽データを検索する手法について検討する．具体的には，学習データに基づくツリーベクトル量子化手法により音楽データをベクトル化し，ベクトル類似度やサポートベクターマシン (SVM) を利用した検索手法の有効性を検証する．また，ユーザからの適合フィードバック情報の適用の効果についても検討を行う．ユーザの主観評価が付与された音楽データに基づいた評価実験の結果，提案手法の有効性が確認された．

Music Information Retrieval Method Based on User Preferences

Keiichiro HOASHI　　Naomi INOUE

KDDI R&D Laboratories, Inc.

2-1-15 Ohara, Kamifukuoka, Saitama 356-8502 JAPAN

E-mail: {hoashi,inoue}@kddilabs.jp

In this research, we propose a music information retrieval method based on user preferences. The method attempts to extract features of music which best express user preferences based on the Tree-based vector quantization (TreeQ) algorithm. Information retrieval algorithms such as vector similarity and support vector machines are applied to retrieve music which the user is expected to prefer. We also propose the implementation of relevance feedback in order to improve retrieval performance, The effectiveness of our methods were proved through evaluation experiments based on a music data collection with user ratings.

1　はじめに

MP3 をはじめとするさまざまなオーディオデータフォーマットの急速な普及にともない，ネット上での音楽情報の流通が盛んになっているほか，個人ユーザでも大量の音楽情報を PC や携帯型音楽再生機に蓄積することが可能になっている．このような背景から，大量の音楽データの中から必要な音楽情報を検索する音楽情報検索システムの必要性が高まっており，近年，音楽情報検索に関連する多くの研究報告が盛んに行われている．

既存の音楽情報検索システムの多くは，検索対象音楽データの中からある特定の楽曲を検索することを目的としている．カラオケシステムなどで曲を検索する場合，一般的には曲のタイトルや演奏アーティスト名など，検索対象曲に付与されている属性データに基づいて検索を行う．一方，音楽データのコンテンツに基づく検索システムもいくつか報告されている．参考文献 [1] などの研究報告では，ハミングによる検索クエリ (query by humming) を入力とし，MIDI データを検索する手法が紹介されている．また，参考文献 [2] では，MIDI のような記号化された音楽データではなく，音楽の実際のコンテンツを検索対象とした音楽情報検索手法について報告されている．

上記の研究などで報告されているような音楽情報検索システムの必要性は今後も高まっていくと考えられる．しかし，このようなシステムは，ユーザが検索したい音楽データについて何らかの事前知識（タイトル，演奏アーティスト，メロディ，等）があることが前提となっており，大量の音楽データの中から未知の音楽情報を発見することは不可能である．そこで，本研究ではユーザの音楽的嗜好を学習し，ユーザが好むと思われる音楽情報を検索する手法について提案する．具体的には，既存のオーディオデータ分類手法に基づき，ユーザの嗜好に基づいた検索手法の有効性を検証する．また，検索精度向上のため，適合フィードバックを適用し，その効果について検証を行う．

2　ツリーベクトル量子化手法

本研究では，Foote が提案したツリーベクトル量子化手法 (TreeQ)[3] に基づき，音楽情報のベクトル化を行う．TreeQ の概要を図 1 に示す．

TreeQ では，ツリー生成の際，対象となる音楽データに正解カテゴリが付与された学習データを利用しており，このカテゴリ情報にしたがって学習データが分類されるよう最適化されたツリーを生成する．次に生成されたツリーに個々の音楽データあるいはカテゴリに属する全音楽データの MFCC のフレームを入力し，そのフレームが到達する葉 (leaf) を求める．そして，各 leaf に到達したフレームの数を算出した結果得られるヒストグラムを対象音楽データまたはカテゴリを表すベクトルとする．

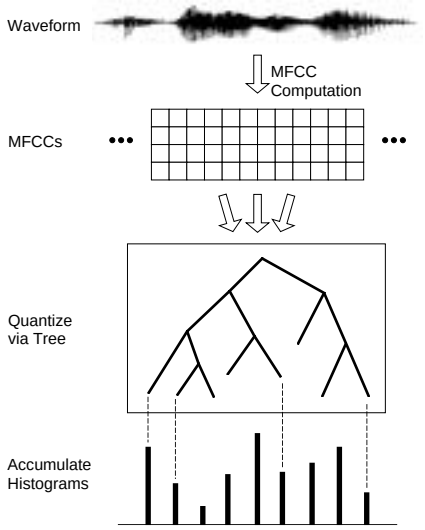

図 1: ツリーベクトル量子化手法概要

Foote の研究報告 [3] では，TreeQ に基づいてベクトル化されたオーディオデータの自動分類手法を提案し，その有効性が示されている．また，Pye らは同手法に基づき，音楽データをジャンル別に分類する手法について報告している [4].

3　問題

TreeQ 手法については，前述の通りオーディオデータや音楽データの自動分類においてその有効性が確認されている．しかし，本手法をそのまま本研究の目的であるユーザの音楽嗜好に基づく音楽情報検索システムに適用するには問題があると考えられる．

最大の問題は，学習データの量である．Foote の研究報告では，学習データとして 255 個のオーディオデータを利用している．また，Pye による音楽のジャンル分類実験では，実験データ 175 個の半分を学習データとして利用し，残りのデータをテストデータとしている．このように，従来研究では数十から数百個もの学習データが必要となっている．したがって，ユーザの音楽的嗜好に基づく検索手法を TreeQ に基づいて実現する場合，ユーザは音楽情報検索の前に大量に好きな曲あるいは嫌いな曲に関する情報をシステムに入力す

る必要がある．しかし，実際に音楽情報検索システムを利用する前に，ユーザが大量の学習データをシステム与えることは非現実的である．そのうえ，音楽的嗜好は音楽ジャンルなどに比べ，曖昧性が高いことが予想され，従来研究よりもさらに学習データ量を増やさない限り，安定して高精度の検索結果を得ることは難しいと推測される．

4 TreeQ に基づく音楽情報検索手法

本研究では，前述の問題について検証するため，まず TreeQ に基づいて検索対象音楽データをベクトル化した上で，以下の 2 つの音楽情報検索手法の評価実験を行う．

1. **ベクトル類似度**
2. **サポートベクターマシン (SVM)**

以上の 2 つの手法について，次節以降で詳しく説明する．

4.1 ベクトル類似度

ベクトル類似度に基づく手法では，TreeQ に基づいてユーザの嗜好を表すベクトルを生成し，検索対象音楽データのベクトルとの類似度に基づいて検索を行う．

まず，学習データとして，ユーザが好む音楽データと好まない音楽データをそれぞれ TreeQ に入力し，ベクトル量子化のためのツリーを生成する（以降，ユーザが好む音楽データの集合を C_g，好まない音楽データの集合を C_b とする）．ツリー生成後，学習データのうち，C_g に属する音楽データをすべてツリーに入力し，ヒストグラムを得ることにより，ユーザが好む音楽データを表すベクトル $\vec{C_g}$ を生成する．また，同様の手法により，ユーザが好まない音楽データを表すベクトル $\vec{C_b}$ も生成する．検索対象音楽データについては，個々の音楽データを 1 つずつツリーに入力することにより，それぞれの音楽データを表すベクトルを生成する．

ここでは，以下の 2 つの数式によってベクトル類似度に基づくスコア算出方法を定義する．

$$Vec_1(K) = Sim(\vec{C_g}, \vec{K}) \quad (1)$$

$$Vec_2(K) = Sim(\vec{C_g}, \vec{K}) - Sim(\vec{C_b}, \vec{K}) \quad (2)$$

ただし，$\vec{K}$ は，検索対象音楽データ K を表すベクトルであるとする．数式 (1) は，単純に $\vec{C_g}$ と $\vec{K}$ とのベクトル類似度のみを算出する方法である．一方，数式 (2) では，$\vec{C_g}$ と $\vec{C_b}$ のそれぞれのベクトルとの類似度を算出し，さらにそれぞれのベクトル類似度間の差を算出することによって検索対象データ K のスコアを得る．この方法は，ユーザが好む音楽との類似度が高く，かつ，ユーザが好まない音楽との類似度が低い音楽データがユーザが好む確率が高いという仮定に基づいており，筆者らの研究 [5] において有効性が確認されている．なお，本研究ではベクトル類似度をコサイン値によって算出する．

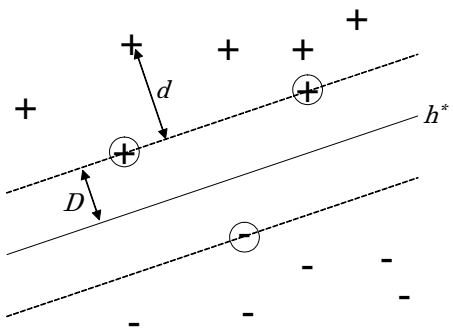

図 2: サポートベクターマシン概要

4.2 サポートベクターマシン

サポートベクターマシン (SVM) とは，主に自動分類やパターン認識などで幅広く利用されているアルゴリズムである [6]．図 2 に SVM の概要を示す．

SVM に基づく音楽情報検索手法では，ベクトル類似度手法と同様，学習データに基づいてツリーを生成する．ツリー生成後，学習データに含まれる個々の音楽データをツリーに入力し，それぞれのベクトルを得る．そして，得られた個々の学習データのベクトルから C_g に属する音楽データと C_b に属するデータを識別するための分離超平面を算出する．

検索対象データのスコアは，ベクトル類似度手法と同様の方法で検索対象データのベクトルを生成し，分離超平面の正例側の境界からの距離（図 2 内の d）を測ることによって算出する．この手法は，分離超平面からの距離が遠い音楽データほど，ユーザが好む確率が高いという仮定に基づいている．SVM は識別器として利用されることが多いが，本手法と同様，分離超平面からの距離に基づいて情報検索を行う手法も報告されている [7]．

表 1: 被験者評価データ

Category	Rating	# of songs	Ratio(%)
C_g	5	1273	14.0
	4	2391	26.4
C_f	3	2798	30.8
C_b	2	1948	21.5
	1	661	7.3

表 2: ベクトル類似度ならびに SVM の検索精度

	Prec@5			Prec@10		
N	Vec_1	Vec_2	SVM	Vec_1	Vec_2	SVM
1	0.5083	0.5183	0.4367	0.4875	0.4867	0.4700
2	0.4967	0.5217	0.4583	0.4825	0.4992	0.4678
3	0.5200	0.5533	0.4967	0.5233	0.5358	0.5167
4	0.4967	0.5217	0.5217	0.4933	0.5042	0.5175
5	0.5383	0.5633	0.5700	0.5092	0.5650	0.5433

5 評価実験

以下，前述の音楽情報検索手法の有効性を検証するために行った評価実験について述べる．

5.1 実験データ

音楽情報検索手法の評価実験用データとして，HMV Japan のウェブページ (`http://www.hmv.co.jp/`) に掲載されている週間 CD 売り上げランキングデータより，2001 年 1 月〜6 月のトップ 10 にランクされた全ての CD アルバム（ベストアルバム，コンピレーションアルバムを除く）60 枚に収録されている楽曲データ 756 件を使用した．

また，ユーザの音楽嗜好のデータを収集するため，実験用データに含まれる全ての楽曲について，被験者 12 名によって 5 段階の主観評価（好き：5〜嫌い：1）が付与された．この主観評価に基づき，それぞれの被験者ごとに実験用データを 3 つのカテゴリ（C_g, C_f, C_b）に分類する．C_g, C_f, C_b には評価値が「4 以上」「3」「2 以下」の音楽データをそれぞれ分類する．全被験者が与えた評価の個数ならびに割合を表 1 に示す．

5.2 実験手法

本実験では，学習用データとして各ユーザの C_g, C_f, C_b からそれぞれランダムに N 件のデータを抽出し，TreeQ に基づいてツリーを生成する．また，検索対象データとして，全実験用データから学習データを除いたデータセットを利用する．学習データのランダム性を考慮し，評価実験は各ユーザについてそれぞれ 10 回ずつ実施する．

5.3 結果

ここでは，ベクトル類似度手法ならびに SVM のそれぞれの手法によって全検索対象データのスコアを算出し，スコア順にソートされた検索対象データ一覧の上位 n 件での検索精度（以下，Prec@n）を算出することにより評価を行う．なお，検索精度算出時には，C_g に含まれるデータを「適合」，C_f, C_b に含まれるデータを「非適合」とそれぞれみなしている．

表 2 に，ベクトル類似度 (Vec_1, Vec_2) ならびに SVM に基づく音楽情報検索手法で得られた全実験結果の検索精度 (Prec@5, Prec@10) の平均値を示す．

表 1 から，全実験データの約 40%が「適合」データであるのに対し，表 2 では全ての手法において 40%を越える検索精度が得られており，TreeQ に基づく音楽情報検索手法の有効性が確認された．また，2 つのベクトル類似度手法 (Vec_1, Vec_2) を比較すると，全般的に Vec_2，すなわち C_g と C_b の両カテゴリベクトルとの類似度を利用した検索手法が特に有効であるという結果が得られた．

一方，SVM とベクトル類似度の結果を比較すると，全般的に SVM の方が検索精度が低い．しかし，学習データ N が増加するにつれ，SVM の検索精度が向上している一方，ベクトル類似度については学習データ量と検索精度の間に明確な相関は見られない．このことから，さらに学習データを増やすことにより，SVM での検索精度が向上する可能性があるといえる．

6 適合フィードバックの導入

前述の評価実験の結果，TreeQ に基づく音楽情報検索手法ではランダムを上回る検索精度が得られた．しかし，検索精度は最大でも約 57%であり，ユーザにとって十分満足できる検索精度には至っていない可能性がある．そこで，本研究では，さらなる検索精度向上を目指し，情報検索において広く利用されている適合フィードバック (relevance feedback) 手法を適用する．以下，ベクトル類似度ならびに SVM に基づく音楽情報検索手法への適合フィードバック (FB) 導入手法について説明する．

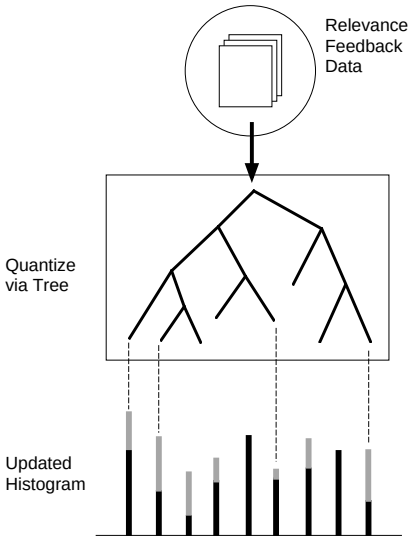

図 3: ベクトル類似度手法における適合 FB

6.1 ベクトル類似度手法における適合 FB

ベクトル類似度手法においては，学習過程で生成されたツリーに適合 FB 情報を入力し，初期カテゴリベクトルにその情報を積み重ねることにより，ベクトルの更新を行う．この手法の概要を図 3 に示す.

図 3 では，適合 FB 情報のうち，適合データの情報を更新前のカテゴリベクトルに蓄積する方法が示されているが，適合 FB に含まれる非適合データについては，適合データとは逆に，カテゴリベクトルの各要素から非適合情報を引くことにより，ベクトル更新を行う．更新前のベクトルを $\vec{C_g} = (c_1, \cdots, c_n)$，更新後のベクトルを $\vec{C'_g} = (c'_1, \cdots, c'_n)$，適合 FB 情報に含まれる適合データのベクトルを $\vec{K_g} = (k_{g1}, \cdots, k_{gn})$，非適合データを $\vec{K_b} = (k_{b1}, \cdots, k_{bn})$ とすると，更新後ベクトルの要素 c'_{gi} は数式 (3) により算出される.

$$c'_{gi} = |c_{gi}| + |k_{gi}| - |k_{bi}| \tag{3}$$

また，上記手法と同様に，$\vec{C_b}$ を更新することも可能である．その場合は，$\vec{C_g}$ の更新とは逆に，適合 FB 情報に含まれる非適合データをベクトルの各要素に加え，適合データを各要素から引くことにより，ベクトル更新を行う．数式 (4) に $\vec{C_b}$ の更新手法を示す.

$$c'_{bi} = |c_{bi}| + |k_{bi}| - |k_{gi}| \tag{4}$$

数式 (3),(4) に示されている適合フィードバック手法は，テキスト情報検索で広く利用されている Rocchio の手法に基づいている [9].

6.2 SVM における適合 FB

一方，SVM に基づく音楽情報検索手法への適合 FB の導入方法は，適合 FB データとして与えられた音楽データを初期の学習データに追加し，改めて分離超平面を算出する方法を採用する．SVM に適合 FB を適用した研究例としては，Joachims の Transductive SVM (TSVM)[8] があげられる．TSVM では，SVM の分別精度を向上させるため，初期学習データで分離超平面を算出後，少量のテストデータの分離超平面からの距離を算出し，分離超平面の正例側の距離が遠いテストデータを正例，残りのテストデータを負例とみなして学習データに追加し，分離超平面の再算出を行う手法である．TSVM は，ユーザから直接 FB 情報を必要としない点では，情報検索で広く利用されている pseudo FB 手法に類似した手法である．これに対し，本研究ではユーザから直接 FB 情報を得る manual FB を想定しているため，前述の手法によって適合 FB を実現する.

6.3 適合 FB 評価実験

以下，適合 FB 手法の評価実験について述べる.

6.3.1 実験手法

適合 FB 情報を得るため，5 節の各実験で得られた検索結果の上位 M 件の音楽データ抽出し，個々のデータが属するカテゴリ (C_g, C_b) を元に各データの適合性を判断し，ベクトル類似度手法の場合は各カテゴリベクトルを更新，SVM の場合は新たに分離超平面を算出する．その結果，得られたベクトルまたは分離超平面に基づき，検索対象データのスコアを再計算し，適合 FB 後の検索結果を得る.

6.3.2 結果

ここでは，5 節の評価実験同様，適合 FB 後の検索結果に対し検索精度を算出する．表 3 に，ベクトル類似度手法ならびに SVM における適合 FB 後の検索精度 (Prec@10) を示す．なお，適合 FB 前の検索結果との公正な比較のため，同表に初期検索の結果から FB に利用した M 件のデータを除いて算出された検索精度も示す（表 3 のカッコ内の値）.

表 3 に示された結果より，ベクトル類似度手法では適合 FB の結果，検索精度が適合 FB 前から全般的に

表 3: 適合 FB 前後のベクトル類似度ならびに SVM の検索精度 (Prec@10)

	Vec_1			Vec_2			SVM		
M	3	5	7	3	5	7	3	5	7
$N=1$	0.4808	0.4767	0.4817	0.4942	0.5067	0.5350	0.4900	0.4925	0.4767
	(0.4675)	(0.4633)	(0.4517)	(0.4775)	(0.4650)	(0.4742)	(0.4683)	(0.4867)	(0.4692)
2	0.4958	0.4883	0.4875	0.5250	0.5342	0.5283	0.4800	0.4858	0.4958
	(0.4783)	(0.4608)	(0.4550)	(0.4792)	(0.4567)	(0.4517)	(0.4608)	(0.4617)	(0.4475)
3	0.5133	0.5367	0.5458	0.5817	0.5858	0.5767	0.5533	0.5592	0.5417
	(0.5067)	(0.4883)	(0.4767)	(0.5283)	(0.5133)	(0.5092)	(0.5200)	(0.5158)	(0.5033)
4	0.5100	0.5167	0.5358	0.5583	0.5717	0.5808	0.5208	0.5200	0.5167
	(0.4875)	(0.4875)	(0.4783)	(0.4983)	(0.4983)	(0.4975)	(0.5083)	(0.5125)	(0.4967)
5	0.5008	0.5017	0.5117	0.5800	0.5800	0.6000	0.5500	0.5642	0.5692
	(0.4967)	(0.4950)	(0.5000)	(0.5475)	(0.5500)	(0.5342)	(0.5383)	(0.5258)	(0.5375)

向上していることが明らかである．特に，Vec_2 の手法では Vec_1 と比較して高い精度向上が得られている．これは，Vec_1 では $\vec{C}_g$ のみを更新しているのに対し，Vec_2 では $\vec{C}_g$ と $\vec{C}_b$ の両方のベクトルを更新しているため，さらに適合 FB の効果が向上したものと考えられる．

一方，SVM でも検索精度が適合 FB 前と比べて向上しているが，ベクトル類似度手法の検索精度と比較した場合，Vec_2 の検索精度を上回る精度は得られていない．その理由として，TSVM では適合 FB として利用したテストデータの一部を適合データとみなし，残りのテストデータを非適合データとみなして十分な FB 情報を得ているのに対し，提案手法では適合 FB として抽出した M 件のデータのみを利用しているため，正負 FB 情報のバランスが悪く，負の FB 情報が不足している可能性がある．したがって，SVM での検索精度を向上させるためには，pseudo FB のような手法によって十分な適合 FB データを得る方法などを検討する必要がある．

7 まとめ

本研究では，ユーザの音楽的嗜好に関する少量の学習データに基づき，ベクトル化された音楽データに対するベクトル類似度ならびに SVM を利用した音楽情報検索手法を提案した．また，両手法に対する適合フィードバックの適用手法を提案し，評価実験を行った．市販 CD から収集した音楽データに対する被験者の評価情報に基づく評価実験の結果，両手法とも高い検索精度が得られ，提案手法の有効性が確認された．また，適合 FB を適用した結果，両手法で適合 FB 前を上回る検索精度が得られた．

謝辞

本研究において多大な貢献をいただいたスウェーデン・Uppsala 大学の Nina Ewerlof 氏に感謝する．This research used the TreeQ package[10] developed by Jonathan T. Foote.

参考文献

[1] Feiten, Gunzel: “Automatic indexing of a sound database using self-organizing neural nets”, Computer Music Journal, 18(3):53-65, 1994.

[2] Shalev-Shwartz, Dubnov, Friedman, Singer: “Robust temporal and spectral modeling for query by melody”, Proceedings of ACM-SIGIR 2002, pp 331-338, 2002.

[3] Foote: “Content-based retrieval of music and audio”, Proceedings of SPIE, Vol 3229, pp 138-147, 1997.

[4] Pye, “Content-based methods for the management of digital music”, Proceedings of ICASSP 2000, Vol IV pp 24-27, 2000.

[5] Hoashi, Zeitler, Inoue: “Implementation of relevance feedback for content-based music retrieval based on user preferences”, Proceedings of ACM-SIGIR 2002, pp 385-386, 2002.

[6] Vapnik: Statistical learning theory, A Wiley-Interscience Publication, 1998.

[7] Joachims, “Optimizing Search Engines Using Clickthrough Data”, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

[8] Joachims, “Transductive Inference for Text Classification using Support Vector Machines”, International Conference of Machine Learning (ICML), 1999.

[9] Rocchio: “Relevance Feedback in Information Retrieval”, in “The SMART Retrieval System – Experiments in Automatic Document Processing”, Prentice Hall Inc., pp 313-323, 1971.

[10] Foote: “The TreeQ Package”, `ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/tools/treeq1.3.tar.gz`